

PLATFORM MANAGEMENT SYSTEM EVOLUTION

Michael S. Tilley

Omlitron, Inc.

6305 Ivy Lane, Suite 500

Greenbelt, Md 20770

(301) 474-1700

INTRODUCTION

This presentation describes a series of studies, performed in fiscal years 1988 and 1989, to derive and refine a concept of evolutionary growth in capability for the PMS.

The presentation is divided into 6 sections:

The first section, PMS History and Background, describes the origin of PMS as an outgrowth of the manned base OMS, describes the differences between platforms and the manned base which led to the necessity for PMS, and shows the distributed nature of PMS within its ground- and space-based context.

The second section describes the seven basic PMS functions and their interactions.

The third section discusses the history of the PMS evolution process, and shows the series of documents which define the baseline PMS and describe its potential evolution paths.

The fourth and fifth sections respectively, discuss the last two documents in the series completed in fiscal year 1989, the Platform Management System Evolution Plan, and the Platform Management System Design Impacts Report.

The sixth section, Conclusions, summarizes the conclusions reached in the above-mentioned documents.

INTRODUCTION

- PRESENTATION CONTENT:**
 - PMS HISTORY AND BACKGROUND**
 - THE 7 BASIC PMS FUNCTIONS**
 - THE PMS EVOLUTION PROCESS**
 - THE PLATFORM MANAGEMENT SYSTEM EVOLUTION PLAN**
 - THE PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT**
 - CONCLUSIONS**

PMS HISTORY AND BACKGROUND

The need for a system to perform operations management functions at the base or platform level (as opposed to the system or payload level) was identified in early 1986. The Operations Management System (OMS) was created to encompass these functions. The platforms are out of ground contact more than the base, and as a consequence, require the platform's operation management function to be more autonomous in order to meet the mission goals of the payloads and platform more effectively. The PMS was split from the OMS in mid-1986 and the OMS was redefined to apply only to the base. The PMS was defined in the latter half of 1986 through a series of meetings attended by people from a wide range of organizations involved with the platform. Operations management functions have typically been performed on the ground and have typically been operator intensive. The functions allocated to the PMS are automated to reduce life cycle costs and are split between space and ground. Three criteria were used to assign functions to the PMS:

- a) Operations management functions that are required to be on-board;
- b) Operations management functions that have the potential to be performed on-board as the platform grows and evolves;
- c) Operations management functions that are directly related to items a and b.

The PMS was defined as the system which coordinates the operation of platform systems and payloads. It provides operations monitoring and control and initiates and maintains autonomous platform operations as required. An important PMS concept is that it is only responsible for operations that occur between the systems and payloads. It is not responsible for operations within a payload or system unless they affect the platform's safety or interfere with another scheduled operation.

The level of automation of the PMS functions is expected to increase over the life cycle of the PMS. Because of limited computing resources on the platforms, and because of limited confidence in the automation of operations management functions, only the essential operations management functions that are required for initial operation, and the stubs for future growth, will be implemented on-board the platform at first launch. This subset of the PMS is called the Platform Management Application (PMA) and resides on the Data Management System's (DMS) computer resources. The balance of the PMS is implemented on the ground in the Platform Support Center (PSC). This ground subset is called the Platform Management Ground Application (PMGA). The PMS will be implemented to allow the smooth migration of functions from the PMGA to the PMA.

Figure 1 shows a block diagram of the distributed PMS.

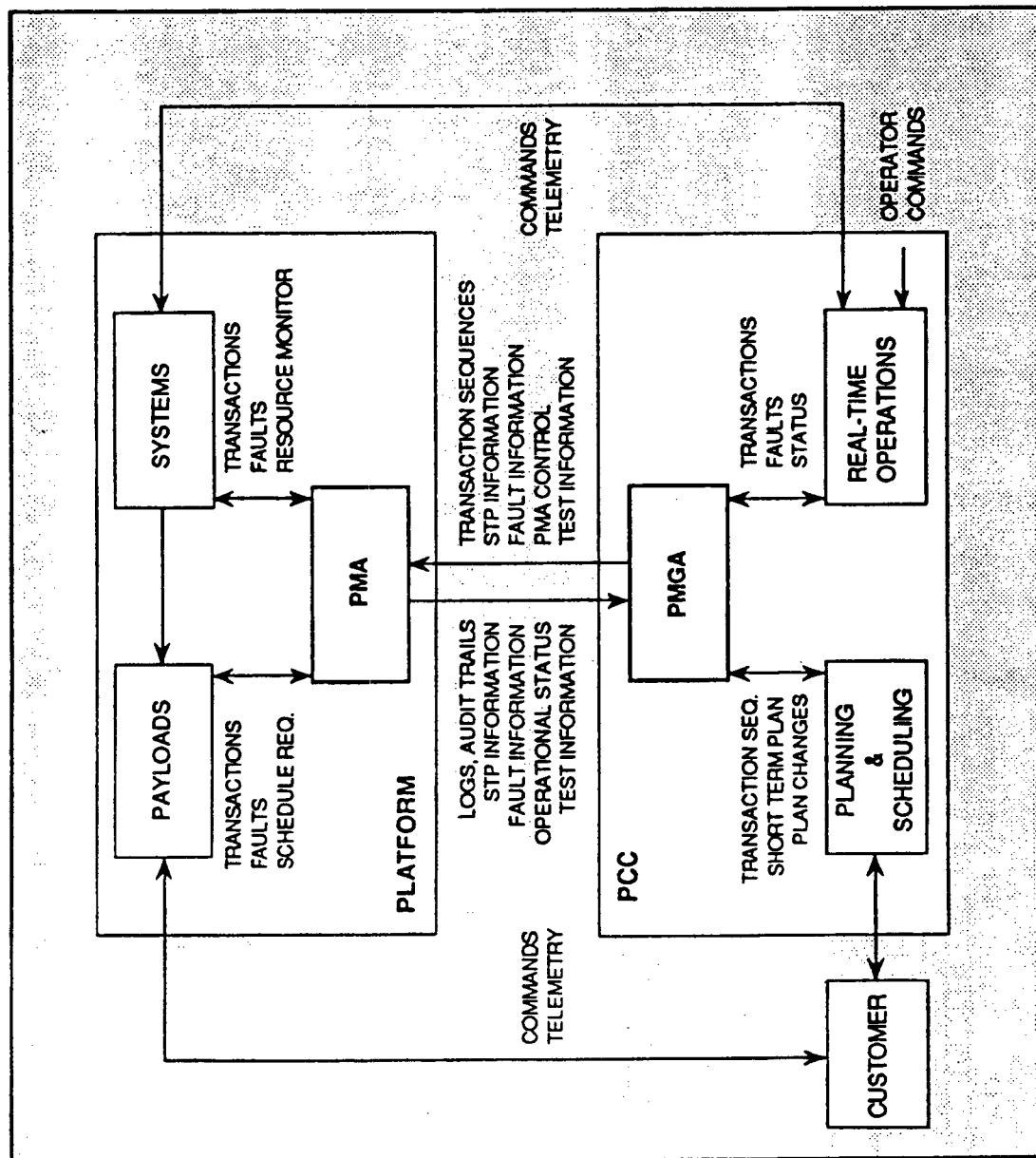


FIGURE 1 PMS DISTRIBUTION

BASIC PMS FUNCTIONS

During the initial definition of PMS, seven basic functions were derived and refined:

Manage the Short Term Plan: The PMS receives a conflict-free Short Term Plan (STP) from the PSC approximately four days ahead of time. The short term plan defines payload and system operation in terms of environmental constraints and resources. The PMS modifies the plan if resource availability changes due to a fault or an incorrect resource prediction. Customers and payloads may request plan modifications (including modifications to ongoing operations). These requests are granted if the resources are available to support the request, or if the priority of the payload and its operation are high enough to displace another operation from the plan.

Manage Operations: The PMS executes the short term plan. It instructs the platform systems on the resources to be supplied to payloads, and controls interlocks on any potentially dangerous payload effectors.

The PMS also provides an optional stored transaction service for customers and operators. Sequences of transactions can be provided to the PMS for execution by time or for execution on the receipt of another transaction. Some transaction sequences may be predefined for use in contingency situations.

Monitor Operations: The PMS receives data from the payloads and systems on resource usage and identifies anomalies. The PMS maintains an audit trail of PMS-related transactions.

Support Testing: The PMS supports on-board interface testing between payloads and/or systems by generating data and logging results.

Recognize and Resolve Conflicts: The PMS recognizes conflicts among the systems and payloads by comparing the resources allocated in the plan and the actual resource availability. The conflict recognition capabilities will be used by the Manage Short Term Plan function in maintaining a conflict-free short term plan. Recognized conflicts will be resolved through a predefined priority scheme.

PLATFORM MANAGEMENT SYSTEM EVOLUTION

Michael S. Tilley

Omlitron, Inc.

6305 Ivy Lane, Suite 500

Greenbelt, Md 20770

(301) 474-1700

ABSTRACT

In fiscal year 1988 a study was initiated to define the Platform Management System (PMS) functions required for the mature platform operations era. The objectives of the task include: 1) defining how to increase the operational productivity of the platform by providing enhanced capability for responding to changing events, 2) influencing the initial PMS design by identifying required hooks and scars and, 3) evaluating potential automation techniques that are appropriate given predicted on-board computing resources.

Initial platform operations scenarios were defined. The focus was on PMS-related functions where operations enhancements are likely to occur. Operations productivity was defined in terms of scientific productivity of the platform as well as the level of automation of the ground system. The Platform Operations Productivity Enhancement Report was completed earlier this year documenting system enhancements to increase science productivity and ground system automation.

Using the baseline PMS defined in the PMS Definition Document as a starting point, the resulting PMS-specific enhancements were molded into a sequence of progressively more sophisticated operations management capabilities. This sequence of upgrades to the PMS has been documented in a PMS Evolution Plan. The plan includes enhancements in the areas of resource scheduling, resource modeling, system and payload anomaly management, and transaction sequence interpretation. A plan for migration of functions from the ground portion of the PMS to the flight portion is also included. The impacts of this plan on the platform are now being documented to ensure that the required hooks and scars are included in the baseline system.

Future plans include a prototype of some of the PMS enhancements to address the feasibility of and techniques for implementing these enhancements in the on-board computing environment.

BASIC PMS FUNCTIONS

- **MANAGE THE SHORT TERM PLAN**
- **MANAGE OPERATIONS**
- **MONITOR OPERATIONS**
- **SUPPORT TESTING**
- **RECOGNIZE AND RESOLVE CONFLICTS**
- **MANAGE INTER-SYSTEM AND PAYLOAD FAULTS**
- **CHECK TRANSACTIONS**

BASIC PMS FUNCTIONS (continued)

Manage Inter-System and Payload Faults: The PMS supervises inter-system and payload fault management and reconfiguration. For certain predefined fault situations, it will support identification of the cause of anomalies and will select an integrated reconfiguration when necessary. If the fault affects the ability of the systems to support the short term plan, the PMS will update the short term plan.

Check Transactions: The PMS will be capable of checking transactions to selected destinations or from selected sources (e.g., PSC operator). Those that are not consistent with the short term plan or safe operation of the platform will not be delivered. The PSC operator can override the check of PSC operator originated transactions.

Figure 2 shows the seven major functions and a greatly simplified view of their interactions.

THE PMS EVOLUTION PROCESS

The series of 4 documents describing the baseline and enhanced PMS are described below, and illustrated in Figure 3.:

PMS Definition Document (October, 1987)

This document described the PMS functions and interfaces. It allocated functions to the space component and to the ground component. This allocation was for the initial platform launch - other ground functions were expected to migrate to the space component as the platforms grow and evolve. The document also contained a set of dataflow diagrams and a data dictionary.

Platform Operations Productivity Enhancement Report (December, 1988)

The PMS described in the PMS Definition Document was a deliberate compromise. On the one hand, it had to support some fairly radical departures (telescience, reactive control) from traditional spacecraft control methods. On the other hand, it had to be kept as simple as possible in recognition of the following:

- the first flight PMS will probably suffer from a severe shortage of computational resources
- the first flight PMS should be simple to minimize operational risk
- as on-board computing resources increase (through later generations of platform, or through servicing of existing platforms) PMS capabilities can be enhanced
- as operational experience is gained, PMS capabilities can be enhanced in areas once considered to be high-risk

In 1988, the Platform Operations Productivity Enhancement Report was delivered. Using the existing PMS definition as a departure point, this document examined ways to increase the operational productivity of the platform, e.g., how to provide the capability to react to changing conditions, such as targets of opportunity and faults, in a manner that comes closer to optimizing the overall platform mission operation. This was achieved by analysis of a number of scenarios, some from flight experience of existing traditional spacecraft, others based on expected platform operations. The results were documented as a series of enhancements to the basic PMS functions.

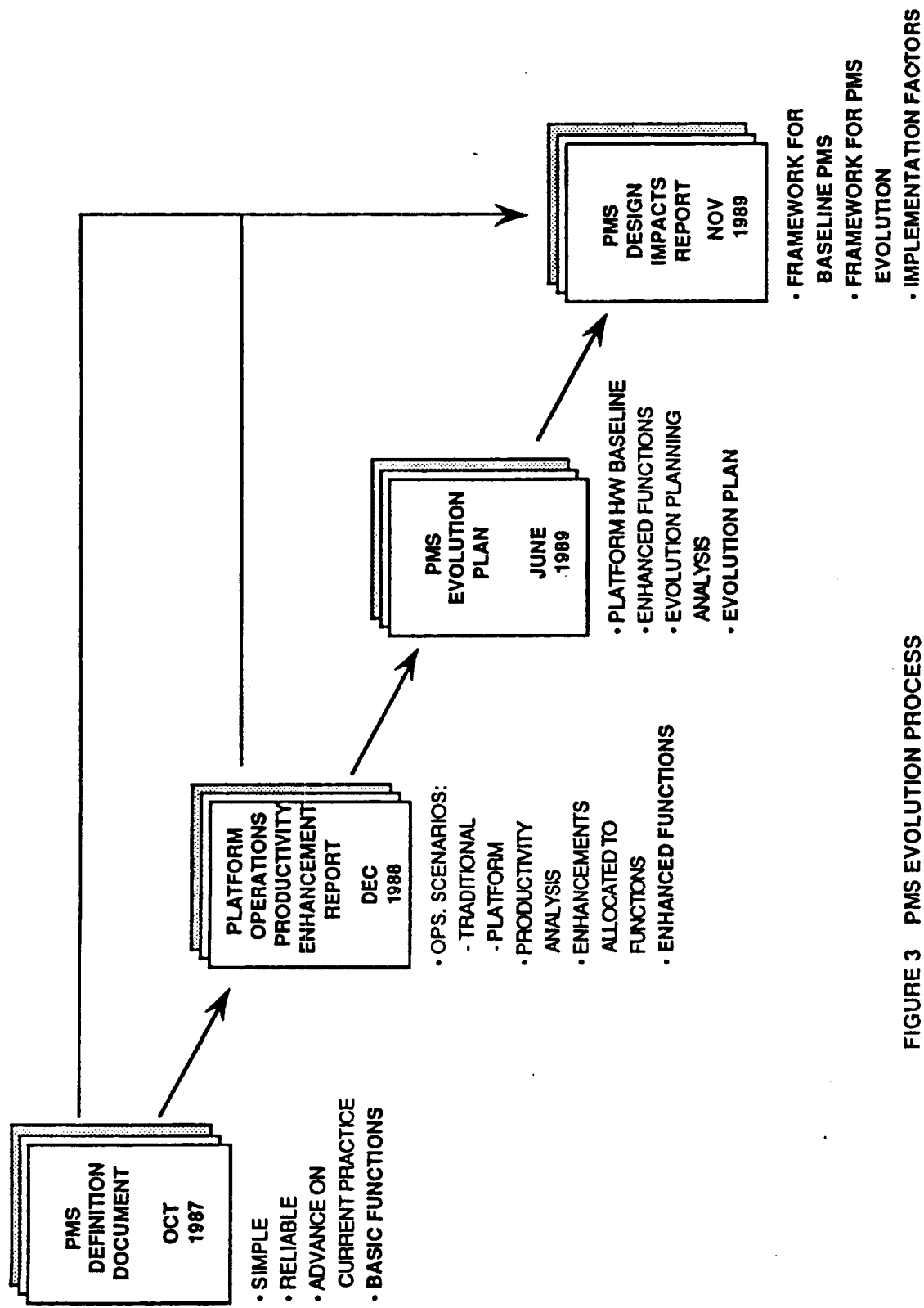


FIGURE 3 PMS EVOLUTION PROCESS

THE PMS EVOLUTION PROCESS (continued)

Platform Management System Evolution Plan (June, 1989)

The purpose of this document was to define a growth path from the baseline Platform Management System to an enhanced PMS based on the Platform Operations Productivity Enhancement Report. The document described the benefits and implementation feasibility of each proposed enhancement, and allocated each enhancement to a PMS evolution phase. It was intended for this document to be used by GSFC Flight Software Systems Branch in monitoring the initial implementation of PMS, and in the maintenance and enhancement of future versions.

Platform Management System Design Impacts Report (to be published November, 1989)

This document, the last in the series, describes the attributes of the Space Station Freedom platform (and its systems) necessary to allow the baseline PMS to function as intended, and notes those attributes of the initial platform which will be necessary to allow PMS evolution. The intended use of the document is to influence the initial design of platform systems to ensure that the infrastructure will exist to allow PMS to fulfill its intended role, and to influence the initial design of PMS itself to ensure that PMS evolution can be achieved at acceptable cost and risk.

PLATFORM MANAGEMENT SYSTEM EVOLUTION PLAN

In considering PMS evolution it is necessary to assume a life-cycle model for the platform hardware and software. While the platform software can be incrementally updated (within the limits of available computing resources), the platform hardware presents only a few discrete opportunities for change - either during on-orbit servicing, or during development of a new platform. Recently, the concept of servicing for polar platforms has been de-emphasized, so that there will be no hardware upgrade during the life of an individual polar platform. However, as new, expendable polar platforms are developed, evolved hardware and software concepts can be applied to them. For co-orbiting platforms, servicing options are still available. Of course, the PMS concept is intended to be applicable to both co-orbiting and polar platforms.

As a device to break up the platform life-cycle into more manageable units, bearing in mind the above discussion, it seems reasonable to suppose the following attributes of a platform life-cycle model:

- first upgrade five years after first launch (either as a result of servicing, or a new platform - current Eos planning involves launching new platforms at 5-year intervals).
- limited hardware changes (except for changes to the payload complement) at the first upgrade.
- second upgrade ten years after first launch.
- potential for extensive hardware change at the second upgrade, for example:
 - replacement of computational resources (standard data processors (SDPs), etc.) with faster devices,
 - addition of more SDPs,
 - additional mass storage.

This model allows three major evolutionary phases for PMS (illustrated in Figure 4):

- | | |
|--------------------|---------------------------------------|
| • BASELINE PMS | from launch to first upgrade. |
| • INTERMEDIATE PMS | from first upgrade to second upgrade, |
| • MATURE PMS | after second upgrade. |

Although useful for planning purposes, the boundary between the phases is somewhat artificial. Flight software can be updated at any time in the life-cycle provided that on-board resources are available.

At any given time, it is quite likely that there will be several platforms in operation at different points in this life-cycle.

The development of hardware and software ground components of the PMS, the Platform Management Ground Application (PMGA), is not constrained to the same degree as the flight hardware and software. However, it is likely that major changes to PMA and PMGA will occur in concert, so it is convenient to use the same life-cycle phases for both.

PLATFORM MANAGEMENT SYSTEM EVOLUTION PLAN

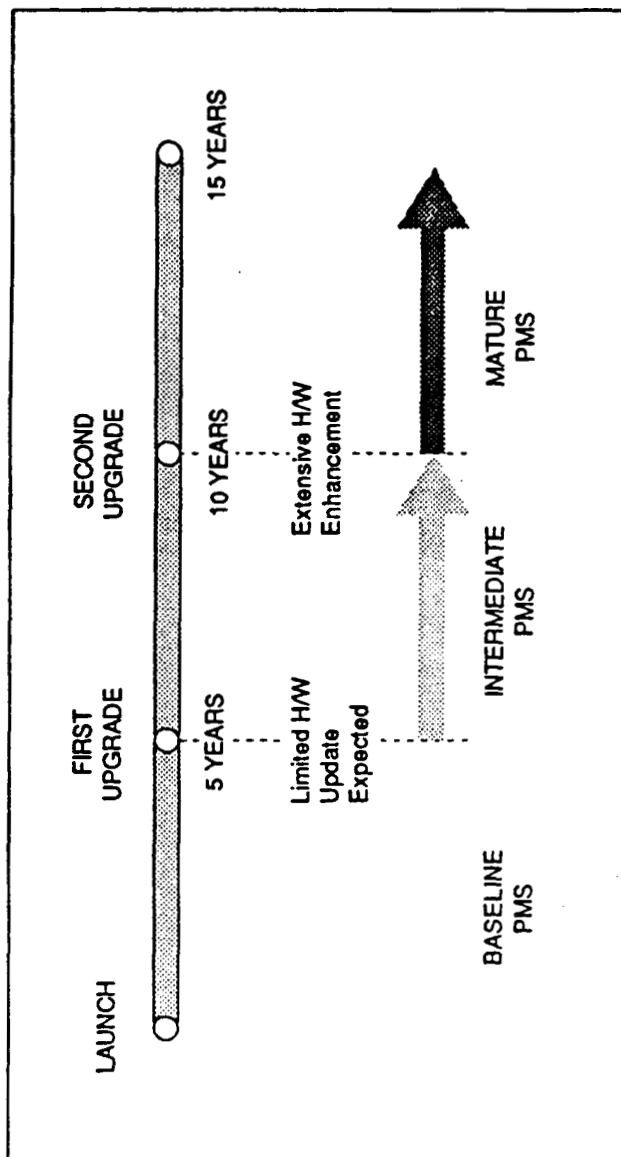


FIGURE 4 PMS EVOLUTION PHASES

PMS ENHANCEMENTS

The PMS Evolution Plan analyzed a number of enhancements proposed in the Platform Operations Productivity Report. These are briefly described below. Each enhancement was allocated a mnemonic identifier (EH-1, etc.) for reference:

EH-1 OBSSTP Integrity Check: Filter the uplinked Short Term Plans (STPs) for elementary errors such as inappropriate time-spans or payload identifiers. The uplinking of incorrect command loads has been a frequent problem on past spacecraft. This is also addressed in EH-5, below. In the PMS era, some of the functions of command loads in traditional spacecraft are embodied in the STP, and so a second aspect of the same problem is the transmission of STP elements and associated transactions to PMS. In this area again, elementary checks performed by PMS could eliminate many problems.

EH-2 Multiple Resource Profiles: Compare the On-Board Segment of the STP (OBSSTP) against several possible resource profiles and choose the best fit, when there is a change in resource availability and a corresponding choice of reconfiguration options. The reconfiguration options are assumed to be provided by the subsystems. A typical situation would be a partial failure of a resource-providing subsystem. Even when autonomous recovery is effected by the failing subsystem, where there is a choice of recovery options, PMA will still be involved. For example, suppose that the Electrical Power System (EPS) has suffered a partial failure and has two alternative recovery actions available - both will result in a reduced power availability profile but, as far as the EPS is concerned, both profiles are equally "good". Clearly PMA will have to determine which profile provides the best fit (i.e., least disruption) to the OBSSTP.

EH-3 OBSSTP trade-offs: Determine the optimum configuration for current OBSSTP when more complex trade-offs are possible. For example, in the case described above, a more favorable (i.e. more productive) power profile might be obtained by changing the platform attitude to gain better solar array coverage, at the expense of some loss of FOV for some instruments. This type of trade-off requires enhancements to several functions. However, recognizing the possibility of alternative strategies (the "what if's) requires an entirely different type of processing ("Artificial Intelligence" for lack of a better term).

PMS ENHANCEMENTS

- OBSSTP INTEGRITY CHECKS (EH-1)**
- MULTIPLE RESOURCE PROFILES (EH-2)**
- OBSSTP TRADE-OFFS (EH-3)**

PMS ENHANCEMENTS (continued)

EH-4 Transaction Interpreter: Implement a transaction interpreter, which would allow a transaction sequence to examine operational conditions, and change execution flow accordingly. This enhancement first surfaced as a possible approach to the provision of a flexible response to faults. The fault recovery of the baseline PMA consists of a combination of hard-coded algorithms and contingency transaction sequences. The nature of a transaction sequence has not been defined in any depth, but seems to be equivalent to traditional spacecraft commands in most respects. Clearly, a predefined command sequence of this type embodies almost no flexibility at execution time. If, however, transaction sequences were expressed in a command language with the ability to interrogate the environment (retrieve engineering values) and execute conditional statements, a great deal more flexibility would be available. This type of approach is used to automate functions in control centers and integration & test systems today. A typical example is the Systems Test and Operations Language (STOL).

EH-5 Integrity Check on Transaction Sequences: Perform basic integrity checks on uplinked transaction sequences. Uplinking of incorrect command loads has been particularly troublesome on past missions. In the PMS era, the baseline PMS can eliminate most of the problems normally caused by human carelessness (it will not forget to send a sequence at the required time; it will not send a sequence with a similar name to the sequence actually required; etc.). The kind of checking required to perform elementary validation of transaction sequences (which should not be confused with Transaction Checking) is not very complicated.

EH-6 OBSSTP screening: When there is a change in resource availability, and a corresponding choice of reconfiguration options, this function will be instrumental in comparing the OBSSTP against several possible resource profiles and choosing the best fit. Note that this enhancement is closely tied to EH-2. The division of labor is as follows:

- EH-2 recognizes the availability of several recovery options.
- EH-10 provides resource availability profiles for each option.
- EH-6 determines which profile has the best fit with the OBSSTP.
- EH-2 chooses the corresponding recovery option.

This enhancement is so closely connected with EH-2 that it is not useful to consider them separately as stand-alone enhancements (they have been described separately up to this point only because of the way they were derived in Reference [3]). For the remainder of this document, we will combine EH-2 and EH-6 into EH-12, Multiple Resource Profiles, and discuss the combination as a single logical enhancement.

PMS ENHANCEMENTS (continued)

• **TRANSACTION INTERPRETER** (EH-4)

• **INTEGRITY CHECK ON TRANSACTION SEQUENCES** (EH-5)

• **OBSSTP SCREENING** (EH-6)

PMS ENHANCEMENTS (continued)

EH-7 Improved Schedule/Reschedule Algorithms: The baseline PMS is handicapped in dealing with many problems because of the lack of scheduling information in the STP. In the event of a schedule conflict, PMA's only recourse is to progressively delete lower priority plan elements until the conflict disappears. This, in itself, will often be a sub-optimal solution. There are two distinct ways in which the situation can be improved. The first involves enhancements to baseline functions to allow plan elements to be moved in time. For example, in the event of an unexpected excess of resources, if the OBSSTP contained the necessary information an evolved PMA could move plan elements forward in time to make use of the available resources. This would not, in itself, improve productivity - the same elements would be executed but at different times. It would, however, improve scheduling flexibility by freeing resources later in the plan. These could then be allocated to payloads making real-time requests, or used in other ways as suggested below. The ability to move elements is also useful in other circumstances. For example, in the event of a resource conflict it might be possible to rearrange the STP elements so that more of them could execute than under the simple deletion scheme of the baseline PMA. This level of enhancement requires changes to PMS to add some genuine rescheduling ability. In addition, considerable information would have to be included in the OBSSTP relating to the scheduling requirements and constraints of each plan element. For example:

- schedule this element only between these times ...
- do not schedule this element while payload X is operating
- this element and element n should be considered as a unit and scheduled together
- do not start this element unless at least x minutes of ground contact remain before the next communication outage.

The second way in which PMA scheduling could be improved is discussed in EH-8, below.

EH-8 Insertion of Unscheduled Events: Support insertion of unscheduled events into the plan. An additional enhancement to PMA's scheduling ability relates to observations which are not contained in OBSSTP at all. In some cases there may be elements of an experimenter's schedule request which were not included in the STP because of resource conflicts or other constraints. If PMA had available a pool of such unscheduled elements, an evolved scheduler could insert them into the plan if the constraints relaxed for any reason. It is also possible that some opportunistic instruments could be routinely operated in this way - scheduled on a "fit this in when you can" basis.

PMS ENHANCEMENTS (continued)

EH-9 Intelligent Anomaly Diagnosis: Intelligent fault recovery requires fault diagnosis and isolation down to the level at which recovery options can be selected, to provide the potential recovery strategies for Conflict Recognition and Resolution to evaluate. This type of goal-directed processing is more usually associated with Artificial Intelligence and Expert System approaches and is difficult to insert into traditionally structured systems. In order to accommodate this enhancement, a different PMS infrastructure might be indicated.

EH-10 Resource Availability Prediction: Maintain (and possibly derive) resource availability predictions. The resource availability profile is of great importance in the recovery from faults, and in scheduling in general. To use a power example again, assume that the EPS has suffered a partial failure and has a choice of two (or more) reconfiguration options. The two choices may result in quite different power availability profiles over the remainder of the orbit and beyond. By itself, the EPS has no way of evaluating the relative merits of these profiles - only PMS can do that with reference to the resource needs of the scheduled operations. Given the availability of predicted resource profiles, PMA could test its OBSSTP against them and use the priorities embedded in the OBSSTP to select the resource profile having the best fit. This function could be relatively trivial (if the subsystems were able to provide their own resource predictions) or quite complex (if it had to model the behavior of one or more subsystems).

EH-11 Manage Payload Anomaly: If we assume that some of the proposed enhancements to PMS involve the use of AI technology, then this implies the existence of an on-board inference-engine of some type and a body of expert knowledge relating to the platform and its subsystems. This body of knowledge could be a set of rules, for example. Under these circumstances, it is conceivable that a set of rules could be supplied for a particular payload. These rules, in conjunction with PMA's inference engine, could act as an "expert advisor" in handling anomalies of the payload. This approach allows PMS a degree of "smartness" about the payload and its activities without compromising its "purity". The storage and use of payload-specific rule sets could be considered to be a PMA service in much the same way as transaction sequences are managed by the baseline PMA.

EH-12 Multiple Resource Profiles: As explained above, EH-12 is a combination of EH-2 and EH-6.

PMS ENHANCEMENTS (continued)

• **IMPROVED SCHEDULE / RESCHEDULE ALGORITHMS (EH-7)**

• **INSERTION OF UNSCHEDULED EVENTS (EH-8)**

PMS ENHANCEMENTS (continued)

- **INTELLIGENT ANOMALY DIAGNOSIS (EH-9)**
- **RESOURCE AVAILABILITY PREDICTION (EH-10)**
- **MANAGE PAYLOAD ANOMALY (EH-11)**
- **MULTIPLE RESOURCE PROFILES (EH-12)**

EVOLUTION PLAN

For each of the enhancements proposed in the previous section, the following topics were examined:

- the benefits of the enhancement to platform and payload operations.
- the feasibility of implementing the enhancement, in terms of:
 - availability of the necessary platform resources,
 - availability of the necessary technology.
- allocation of the enhancement to one of the PMS evolution phases discussed above:
 - PMA/PMGA baseline (can and should be added to baseline).
 - PMA/PMGA intermediate (1st major s/w upgrade, h/w unchanged).
 - PMA/PMGA mature (2nd major s/w upgrade, possible h/w upgrade).

In addition to the factors listed above, the allocation to PMS evolution phases also considered the dependencies between the proposed enhancements, since few of them are completely stand-alone.

Finally, the document summarized the time-phased implementation of the enhancements. These are illustrated in Figures 5 and 6 (the phasing of EH-10 is shown separately in Figure 6 for clarity). The migration of functions from PMGA to PMA is also shown. For the sake of discussion, the time-phasing corresponds to the evolution phases already presented, however, with respect to software changes, there is no reason to artificially restrict updates to predetermined servicing epochs. Software changes that are essentially stand-alone may be implemented or migrated in a continuous fashion, with due consideration for platform resources and ongoing platform operations.

EVOLUTION PLAN (continued)

Enhancement EH-10 (Resource Availability Prediction) has a wide range of implementation options depending on the capabilities of individual subsystems to predict their own resource availability profiles. To avoid too much clutter on Figure 5, the evolution and migration phasing of EH-10 is shown separately in Figure 6.

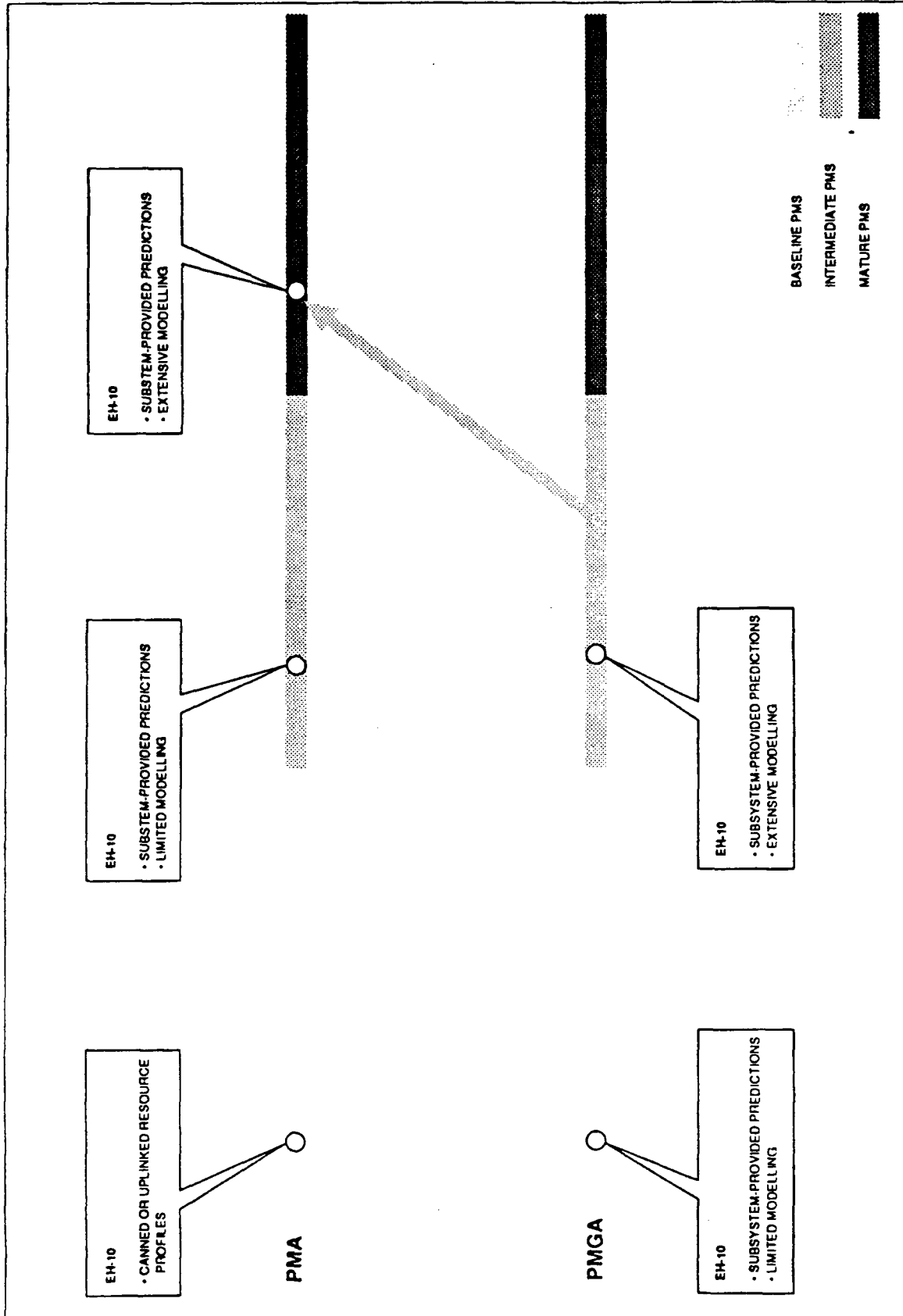


FIGURE 6 EH-10 (RESOURCE AVAILABILITY PREDICTION) EVOLUTION AND MIGRATION PHASING

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT

The PMS Design Impacts Report consists of 3 main sections. The first considers the platform hardware and software attributes required by the baseline PMS. It is divided into the following sub-sections corresponding to the platform resource providers:

- Data Management System (DMS)
- Communication and Tracking (C&T) System
- Electrical Power System (EPS)
- Guidance, Navigation and Control (GN&C) System
- Thermal Control System (TCS)

The second section considers the platform hardware and software attributes required to allow PMS to evolve into the more advanced system described in the PMS Evolution Plan. The opening sub-section discusses the required evolutionary attributes of the baseline PMS itself. The remaining sub-sections are concerned with the major platform systems (as above).

In the previous sections (and also in the preceding documents) the PMS baseline and enhanced functions were discussed in terms of their usefulness, without much concern over how they could be implemented. The third section of the Design Impacts Report examines the potential difficulties in PMS implementation, looks at work in progress to alleviate some of these difficulties, and suggests some additional studies. This section is divided into sub-sections corresponding to a natural grouping of PMS implementation issues into the following areas of implementation risk:

- flight CPU.
- scheduling.
- artificial intelligence.
- subsystem modelling.
- on-board transaction interpreter.

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT

- THREE MAJOR SECTIONS:

- PLATFORM H/W AND S/W ATTRIBUTES REQUIRED TO IMPLEMENT BASELINE

PMS

- PLATFORM H/W AND S/W ATTRIBUTES REQUIRED FOR PMS EVOLUTION
- POTENTIAL IMPLEMENTATION PROBLEMS AND SUGGESTED FURTHER WORK

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - BASELINE H/W & S/W

REQUIREMENTS

The document lists a number of requirements on the platform systems which are required for the implementation of the baseline PMS. It is not useful to reproduce the requirements here, however, the document lists requirements in the following areas:

- on-board CPU
- local area networks
- database and file management
- tape-recorders
- guidance, navigation and control
- system services
- mass storage
- downlink
- electrical power
- thermal control

In addition to the specific requirements as mentioned above, there are several general requirements applicable to all subsystems:

- each subsystem should be able to measure and control the resource(s) it provides
- each subsystem must allow PMS to specify the level of its resource to be supplied to each payload
- each subsystem should support fault isolation and recovery

Finally, several requirements for on-board tape recorder (T/R) control are of particular note:

- In order for PMA to monitor usage of the tape recorder resource, PMA requires the T/R system to:
 - measure incoming bit rates from each payload
 - report these usage rates to the PMA
- In order to allow rescheduling of resources by the PMA in response to faults or requests from users, the T/R software must have a tape recorder scheduling function which:
 - receives a tape recorder resource envelope at time t1 from the PMA which specifies both the planned ingest rates (in bps) for each of the payloads and the downlink bandwidth available at time t2,
 - calculates the best configuration of operating tape recorders, record/playback speeds, and bitstream switching to support the envelope at time t2,
 - generates the appropriate tape recorder and data stream switching commands between time t1 and t2 to effect the new T/R configuration.

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - BASELINE H/W & S/W

REQUIREMENTS

• BASIC REQUIREMENTS FOR:

- ON-BOARD CPU
- SYSTEM SERVICES
- LOCAL AREA NETWORKS
- MASS STORAGE
- DATABASE AND FILE MANAGEMENT
- DOWNLINK
- TAPE-RECORDERS
- ELECTRICAL POWER
- GUIDANCE, NAVIGATION AND CONTROL
- THERMAL CONTROL

• GENERAL REQUIREMENTS FOR:

- RESOURCE MEASUREMENT AND CONTROL
- FAULT ISOLATION AND RECOVERY

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - EVOLUTION

REQUIREMENTS

Very few evolutionary requirements on platform subsystems were identified. As in the case of the baseline PMS, the limits on PMS evolution will ultimately be decided by the on-board computing infrastructure (CPUs, LANs, mass storage, etc.) more than any other factor. Unless there is flexibility and transparent extensibility in these areas, evolution will be difficult or impossible.

Although the platform environment is important in determining the limits of PMS evolution, the baseline PMS itself needs to be implemented in a way that will permit both evolution of functions, and migration of functions from the ground:

- PMS design needs to stress modularity with particular emphasis on defining function boundaries such that, as far as possible, the effects of any evolutionary change are confined to a single function.
- To facilitate function migration from ground to flight segments, PMS design needs to stress transportability. Originally, this was mandated by Space Station Freedom policy (for example, the use of a common, transportable operating system for both flight and ground elements). Although the polar platform is no longer bound by this mandate, the potential benefits should not be discarded hastily.
- Several proposed enhancements require additional information to be present in the STP. Design of the STP should be such that its structure can be changed without requiring code changes to the flight software. Changes to flight code will then only be required if new STP fields are added, or if the meaning of an existing field is changed.
- Several proposed enhancements have a significant Artificial Intelligence (AI) content. Rather than attempt to add these to an existing, traditional program structure, it would be beneficial to consider designing PMS from the outset as a distributed, extensible AI system.

**PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - EVOLUTION
REQUIREMENTS**

- MAJOR EVOLUTION REQUIREMENT ON PLATFORM IS FLEXIBILITY AND EXTENSIBILITY OF
COMPUTING INFRASTRUCTURE (CPUs, LANs, MASS STORAGE, ETC.)
- THE BASELINE PMS ITSELF NEEDS TO BE DESIGNED FROM THE OUTSET WITH
MIGRATION AND EVOLUTION IN MIND

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

FLIGHT CPU

As mentioned above, PMS is a software system and, like any software system, is potentially constrained by the CPU on which it executes. In the case of PMS, the situation is complicated by a number of unknown factors:

- the amount of on-board CPU-time available to PMS is unknown,
- the amount of processing that can be done with the available CPU-time (i.e., the speed of the CPU) is undecided,
- the amount of on-board RAM available to PMS is unknown.

Given the original Space Station Freedom (SSF) platform architecture, none of these represented insurmountable problems. The CPU was fast enough and expandable enough that, despite the uncertainties over the available CPU budget, implementation of the baseline PMS seemed very plausible. It was also clear that the budget would be exceeded by the proposed evolutionary enhancements. However, since there was a phased implementation strategy for the PMS enhancements, and a clear growth path for the CPU, the evolutionary enhancements also appeared plausible.

Recently, with the separation of the polar platform from the Space Station Freedom Program, the choice of on-board CPU has again been thrown open. While this creates additional uncertainties, there are two ways that the situation could be improved:

- The PMS functions should be modelled and prototyped to establish realistic CPU and RAM requirements. Some of this has already been undertaken in the PMS testbed for the baseline functions. Further prototyping of the enhancements should be carried out to provide the basis for a realistic implementation plan.
- The choice of the on-board CPU should be made very carefully with regard to an available growth path. This is an issue not just for PMS, but for the evolution of the platform as a whole. The SSF CPU (Intel 386) has a very clear growth path to the 486 (already shipping), the 586 (already produced in small quantities) and beyond. This continuing growth in CPU capability and reduction in power consumption is driven by relentless commercial pressures. The benefits are available to NASA at practically zero cost. The trade-off between these benefits and the problems of radiation hardening (which will probably never be addressed by the commercial sector) must be made very cautiously. An inappropriate choice at this point could restrict a whole generation of platforms.

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

- FLIGHT CPU**
 - PMS CPU BUDGET**
 - FLIGHT CPU PERFORMANCE**
 - RAM AVAILABILITY**
 - CHOICE OF CPU ON REDESIGNED PLATFORM**

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

(continued)

SCHEDULING

The very simple rescheduling capability of the baseline PMS poses no great implementation challenges. However, several of the proposed enhancements involve more sophisticated methodologies which pose a greater implementation risk. There are two distinct challenges. The first is that 'smart' schedulers tend to be resource intensive which, in turn, becomes another facet of the CPU/RAM budget problem discussed above. The second challenge is that of scheduling technology itself. Although a lot of work is being done in this area, the platform environment imposes some unusual constraints:

- 'smart' schedulers have not been tried in autonomous flight systems.
- there is not much experience with schedulers operating in real-time.

These concerns have been at least partially addressed in two ways:

- Computer Sciences Corporation has been working with GSFC Code 520 on an automated scheduling task funded by the SSF Transition Definition Program. This task has implemented a prototype PMGA and PMA scheduler running on an IBM PC computer. This prototype demonstrates real-time rescheduling in response to events, with synchronization of the schedulers taking place during simulated ground contacts. In addition to exploring interactions between the schedulers, the prototype demonstrates real-time conflict recognition and resolution. The schedulers are based on the PMS baseline definition with an additional capability of adding new elements in response to an increase in available resources (a similar capability is one of the proposed PMS enhancements).
- General Electric (GE)Astro Space Division proposed a scheduler for PMS, having many of the enhanced capabilities, based on an existing PC-hosted Landsat scheduler. The current status of this project is unknown.

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

(continued)

• SCHEDULING

- SMART SCHEDULERS

- SCHEDULING IN REAL-TIME ENVIRONMENT

- CODE 520 AUTOMATED SCHEDULING TASK

- GE LANDSAT SCHEDULER

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

(continued)

ARTIFICIAL INTELLIGENCE

Several of the proposed PMS enhancements identified a need for problem solving of the type often ascribed to artificial intelligence or expert systems, particularly in the area of anomaly diagnosis. As in the previous section, this poses challenges because of the limited CPU availability, and also because the technology is somewhat unexplored, especially in flight systems.

Another GE proposal for the SSF platform was an Expert System for Platform Anomaly Diagnosis (ESPAD). In addition to meeting the requirements of one of the proposed PMS enhancements (EH-9), it appeared that several other enhancements could be achieved via extensions to the ESPAD rule-base. A prototype of the inference engine exists and has returned promising benchmark results. Once again, the current status of this project is uncertain.

SUBSYSTEM MODELLING

One of the things that emerged from the Platform Operations Productivity Enhancement work is the importance of good resource predictions in enabling an advanced scheduler to make good scheduling decisions. This involves modelling the behavior of subsystems to a reasonable degree of fidelity, and could be done by the subsystem, or by PMS, or a mixture of both. While this has been done in ground engineering systems it is, once again, a novel application for flight systems. This type of modelling is a good candidate for migration from ground to flight systems. It is also possible that significant results could be obtained using a rule-based system such as ESPAD.

ON-BOARD TRANSACTION INTERPRETER

The proposed on-board transaction interpreter is unusual when compared to the other enhancements, in that it does not require large amounts of CPU resources, and the technology is very well understood from extensive ground-system experience. However, this enhancement did arouse some concern over the possible operational impacts. It might be fruitful to prototype the transaction interpreter, not to explore the technology, but to demonstrate the potential operational benefits.

PLATFORM MANAGEMENT SYSTEM DESIGN IMPACTS REPORT - IMPLEMENTATION ISSUES

(continued)

- **ARTIFICIAL INTELLIGENCE**
- **SUBSYSTEM MODELLING**
- **ON-BOARD TRANSACTION INTERPRETER**

CONCLUSIONS

The PMS definition and evolution documents described above have demonstrated a number of important conclusions:

- Implementation of a Platform Management System does involve more risk than a traditional spacecraft on-board system, however, the potential benefits are numerous:
 - greater flexibility in platform operations,
 - better response to unplanned events (targets of opportunity),
 - less observation time lost in case of recoverable faults,
 - increased science data return,
 - more efficient use of platform resources
 - reduced costs for ground operations
 - directly supports improved user/instrument interaction (telescience).
- In order to achieve these benefits, certain characteristics of the platform and its subsystems are required:
 - powerful on-board CPU(s),
 - flexible on-board communications (LANs),
 - resource-providing subsystems able to measure and control the resources they provide
 - an architecture that supports incremental and transparent growth in on-board computational capability.

Given the above characteristics, it is possible to implement a platform which is more capable than traditional systems by orders of magnitude, and which can provide a suitable foundation for years of future growth.

ACKNOWLEDGEMENTS

The PMS evolution effort was funded by the Office of Space Station Strategic Plans and Programs Division, and managed by Goddard Space Flight Center Flight Software Systems Branch.

REFERENCES

1. "Space Station Operations Task Force Summary Report," NASA, October, 1987.
2. Fishtahler, L. / Tilley, M. / Tompkins, S., "Platform Management System (PMS) Definition Document," GSFC, October, 1987.
3. Hawkins, F. / Simm, D. / Tilley, M., "Platform Operations Productivity Enhancement Report," GSFC, December, 1988.
4. Stakem, P. / Tilley, M., "Platform Management System Evolution Plan," GSFC, June, 1989.
5. Hartley, J. / Tilley, M., "Platform Management System Design Impacts Report," GSFC, November, 1989.